

CAN (Controller Area Network) のためのマルチメディア 伝送プロトコルの設計と評価

佐藤 健哉[†] 井上 博之[‡]

組み込みリアルタイムシステムは、自動車、家電製品、産業用機器など幅広い分野に適用されている。CAN (Controller Area Network) は、自動車用途に開発され、高速応答性、高信頼性が重要なアプリケーション分野におけるセンサーやアクチュエータを制御するための、標準データリンクプロトコルとして利用されている。本研究では、CANを利用してマルチメディアアプリケーションを用意を実現するための論理的通信モデルと優先度制御をサポートしたデータ転送プロトコルの設計・評価を行った。この通信モデルとデータ転送プロトコルを利用することで、下位のネットワークをほとんど意識することなく効率的に組み込みシステム向けアプリケーションを実現することができる。CAN仕様のデータフレームは、29ビットのヘッダと最大8バイトのデータから構成され、上位のプロトコルを構築するには制約の大きい構造となっているため、本論文では、データ転送プロトコルを効率的にCANのデータフレームへのマッピングする方法も記述した。

Implementation and Evaluation of Multimedia Transmission Protocol on CAN (Controller Area Network)

KENYA SATO[†], and HIROYUKI INOUE[‡]

Embedded real-time systems are deployed in a wide range of application domains such as transportation systems, automated manufacturing, home appliances and telecommunications. Originally developed for use in automotive applications, CAN (Controller Area Network) has been adopted by industry as the standard network technology to transmit data for sensors and actuators, due to its fast response and high reliability for applications. We design and implement a logical communication model and a real-time data transmission protocol for embedded systems with CAN. This logical communication model with prioritized data transmission is effective for developing application programs in an embedded system, which results in less consideration of underlying network mechanisms and the high portability of applications. Owing to the small size of the CAN frame format, there are many restrictions on the implementation of a network protocol on CAN. In an attempt to solve this problem, we describe a concrete implementation of the network protocols in detail to realize the logical communication model a CAN 8-byte data frame with a 29-bit identifier.

1. はじめに

自動車、家電製品、産業用機器など幅広い分野において組み込みリアルタイムシステムが適用されている。CAN (Controller Area Network) [1] [2] は、自動車用途に開発され、高速応答性、高信頼性が重要なアプリケーション分野におけるセンサーやアクチュエータを制御するためのネットワークの標準として利用されている。一般に、組み込みシステムは多様多様であるが、半導体技術の進歩とデジタル化、ネ

트워크化に伴い、その構成は複雑になり、高度なものは一種の分散システムの様相を呈している。これに伴い、ソフトウェア設計も大規模、複雑化する傾向にあり、モジュール化、コンポーネント化を行うことで、開発工程の短縮、信頼性の向上が行われている。

本研究では、マルチメディアを対象とした組み込みシステム向けの論理的通信モデルを構築し、CANのデータリンク層の上にこのモデルを実現するための通信プロトコルを設計し、実装と評価を行う。この通信モデルでは、アプリケーションの概念として、論理ユニットを定義し、論理ユニットの識別子をもとに、アプリケーションが通信を行う形態を採る。そのため、アプリケーションは、どの機器(物理ユニット)に搭載されていても、それを意識することなく通信を行うことが可能となる。CANのフレームフォーマットは29ビット

[†] 同志社大学 工学部 情報システムデザイン学科
Department of Information Systems Design, Doshisha University
[‡] (株)インターネット総合研究所 ユビキタス研究所
Ubiquitous Laboratories Dept., Internet Research Institute, Inc.

のIDと最長8バイトのデータで構成されるため、一般的に、仕様の複雑なプロトコルをこの上に構成することは困難である。そのため、上位のプロトコル仕様をコンパクトにし、CANのフレームフォーマットの上に効率よく実装することが必要となる。本報告では、この論理的通信モデルをCAN上に実装するための、具体的なフレームフォーマットについても記述する。

2. 関連研究

自動車用マルチメディア、テレマティクスのアプリケーションとして、ITSデータベース(IDB)が検討されている。図1にIDBのネットワーク構成を示す。自動車メーカー独自のバスとIDBがゲートウェイで接続されており、IDBのネットワーク側にIDBデバイスが接続される。IDBデバイスは、標準化されたバスインタフェースを持つため、どの自動車メーカーのネットワークに対しても接続が可能となる。自動車メーカー独自のバスは、自動車メーカーごとに仕様異なるが、ゲートウェイを通して、IDB側から標準化された手順でアクセスすることが可能となっている。

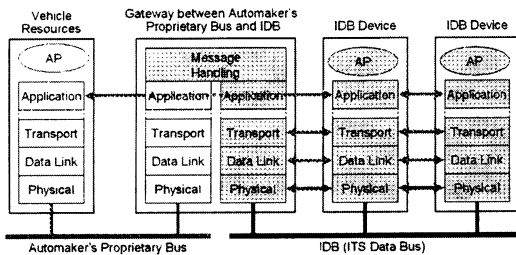


図1: IDB ネットワークモデル

このIDBをCAN上で実現した仕様 がIDB-Cである。IDB-Cは、SAE (Society of Automotive Engineering) において、物理層 (J2366-1) [3], データリンク層 (J2366-2) [4], トランスポート層 (J2366-4) [5], アプリケーション層 (J2366-7) [6]からが標準化されているが、(1) レスponse時間が長い、(2) データ転送効率が悪い、(3) CPUの負荷が高い、(4) 論理的識別子を利用したアプリケーションの構成が困難、といった欠点がある。

CANのMAC副層は、CSMA/CR (Carrier Sense Multiple Access / Collision Resolution)を採用しており、通常システムではCANコントローラのハードウェアにおいてメディアアクセス制御を行うため、CPU負荷が小さく、低レイテンシー、高スループットを実現している。しかし、IDB-Cのデータリンク層では、このCSMA/CRの上位にトークンパッシングを構築している。これはメディアアクセス制御として冗長であり、最大遅延が10msとレイテンシーの増加の要因となる。

また、CANのスループットの約1/3をこのトークンパッシングに利用するため、低スループットとなる。さらに、IDB-Cフレームは通常CANコントローラで処理できず、不必要なフレームであってもCPUで処理するため、CPUの負荷が高くなる要因となる。IDB-Cのデータリンク層では、後述する論理的通信モデルを実現する機能がないため、アプリケーションの設計を困難にし、また、ゲートウェイの処理を複雑にする要因ともなる。

3. 論理的通信モデル

組み込みシステムにおける論理的通信モデルでは、物理的コンポーネント内にある論理識別子により、他のユニット内のアプリケーションと、直接通信を行うことができる。それぞれのアプリケーションは、搭載されているコンポーネント、あるいは、下位のネットワークの階層を意識することなく設計が可能となり、また、その機能を容易に他のユニットに移動させることもできる。また、論理識別子を利用すれば、車輪速センサー、エアバッグ用Gセンサー、ドアロックアクチュエータなどといったサービスの種類を特定させることにも利用でき、特別なサービスディスカバリ機構がなくても、アプリケーションはサービスを利用することが可能となる。

3.1 論理ユニット

図2に示すように、論理的通信モデルにおいて、アプリケーションモジュールの抽象化を論理ユニット、物理的コンポーネントを物理ユニットと定義する。論理ユニットはその論理識別子としての論理アドレスを持ち、この論理アドレスは、物理ユニットの内部で一意に定義される。物理ユニットは物理的な識別子として物理アドレスを持つ。

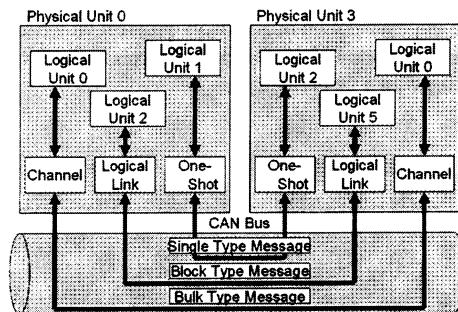


図2: 論理的通信モデルの構成

この例では、1つの物理ユニット内に3つの論理ユニットが存在する。2つの物理ユニットはCANで接続されている。物理ユニット0内には、論理ユニット0, 1, 2があり、物理ユニット3内には論理ユニット0, 2, 5があ

る。物理ユニット0内の論理ユニット0は、物理ユニット3内の論理ユニット0と、仮想的に接続されており、逐一、相手方の物理アドレスを確認することなく通信することが可能である。

論理的通信モデルを用いてメッセージの優先度制御を実現するためには、アプリケーション層の通信プロトコルばかりでなく、トランスポート層やデータリンク層のプロトコルにおいても、論理ユニット間の仮想的通信を認識し、制御する必要がある。各階層における通信の詳細については、4章、5章で述べる。

3.2 伝送方式

組み込みリアルタイムシステムでは、ネットワーク上を伝送するメッセージの緊急性がそれぞれ異なり、緊急性の高いメッセージは小さな遅延で伝送されることが求められる。たとえば、自動車のアプリケーションにおいては、ハンドル操作、アンチロックブレーキ、エアバッグなどの情報を送るメッセージは、他のメッセージ、たとえば、窓の開閉や、ヘッドライトのオン・オフを制御するためのメッセージと比較して緊急性が高いと言える。また、運転記録や故障診断のログデータなどは緊急に転送する必要がないが、メッセージのサイズが大きくなるため、効率的に伝送することが求められる。したがって、メッセージの伝送は、アプリケーションごとに要求が異なり、通信の機構において、異なる要求のメッセージを分類して低レイテンシー、高スループットをサポートする必要がある。

図2に示すように、本研究における通信モデルでは、メッセージの伝送を、それぞれの要求に応じて、「シングル」、「ブロック」、「バルク」の3つのタイプに分類し、それぞれのタイプを伝送する通信路を、「ワンショット」コネクション、「論理リンク」コネクション、「チャンネル」コネクションと呼ぶ。

ワンショットコネクションを通るシングルタイプメッセージは、メッセージ長が小さく、最も高い優先度を持ち、最小のレイテンシーで伝送される。論理リンクコネクションを通るブロックメッセージは、中程度の優先度で、中程度のレイテンシーを伝送するメッセージである。チャンネルコネクションを通るバルクメッセージは、優先度が低くレイテンシーは大きい、効率良くメッセージを伝送することができる。また、チャンネルごとに優先度を設定することができる。1つの論理ユニットが異なった種類のコネクションを同時に設定することも可能である。

4. データリンク層

ここでは、論理的通信モデルにおいて、優先度を持ったメッセージの伝送を実現するためのCAN上に構築するデータリンクプロトコルについて解説する。

29ビットIDと最長8バイトのデータからなるCANのフレームフォーマットに効率良く機能を設定することが重要となる。MAC副層としてCANのCSMA/CRをそのまま利用し、LLC副層として優先度を持ったメッセージを伝送するための論理的通信モデルをサポートする機能を搭載している。

4.1 メッセージタイプ

図3にデータリンク層のフレームフォーマットを示す。

MAC Type	Destination Physical Address	Source Physical Address	Command Type	MAC Data	
Single Type	Destination Physical Address	Source Physical Address	Destination Logical Address	Source Logical Address	Message Control
Block Type	Destination Physical Address	Source Physical Address	Logical Link ID	Counter	Message Control
Bulk Type	Destination Physical Address	Priority	Channel ID	Counter	Message Control

図3: データリンク層フレームフォーマット

フレームフォーマットにおいて、(1) MACタイプ、(2) シングルタイプ、(3) ブロックタイプ、(4) バルクタイプの4つのタイプを定義する。MACタイプは、送信先物理アドレス、送信元物理アドレス、コマンドタイプ、MACデータフィールドからなる。MACタイプメッセージは、物理ユニットの制御、メッセージのACK/NACKに利用され、論理アドレスを含まない。ブロックタイプは、送信先物理アドレス、送信元物理アドレス、送信先論理アドレス、送信元論理アドレス、メッセージコントロールフィールドからなる。論理的通信モデルをサポートするため、データリンク層のメッセージフォーマットに論理アドレスをフレーム内に設定している。バルクタイプは、送信先物理アドレス、優先度情報、チャンネルID、カウンタ、メッセージコントロールフィールドからなる。論理リンクは送信元、送信先の物理ユニット間で設定されるが、チャンネルは送信先の物理ユニットごとに設定される。それぞれの識別子は物理ユニットで設定されているが、どちらのコネクションも論理ユニット同士を接続するためのものである。

4.2 メッセージの優先度

メッセージの優先度は、MACタイプ、シングルタイプ、ブロックタイプ、バルクタイプの順で設定される。これにより、たとえば、長いバルクタイプメッセージ伝送中であっても、シングルタイプメッセージを送信することが可能となる。ブロックタイプより、バルクタイプのカウンタが大きく、長いメッセージを効率的に送ることができる。

図4にシングルタイプメッセージ(ワンショットコネ

ション), ブロックタイプメッセージ(論理リンクコネクション), バルクメッセージ(チャンネルコネクション)について, それぞれの優先度をもったメッセージの送信モデルを示す。

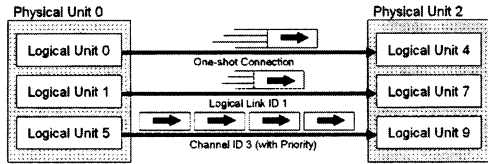


図 4: メッセージタイプの優先度

シングルタイプメッセージは, 送信元と送信先の論理アドレス間で設定され, 論理リンク識別子, チャンネル識別子のコネクションを示す情報を持たない。図 4 の例では, 物理ユニット0内の論理ユニット0から, 物理ユニット2内の論理ユニット4へメッセージが送信される。この送信元, 送信先の論理ユニット識別子は, メッセージ内に含まれる。トランスポート層のACK, NACKのメッセージは, シングルタイプメッセージを利用する。ブロックメッセージは, 論理リンクを利用して送信先の論理ユニットを指定するが, 論理リンクは, 送信元物理アドレス, 送信先物理アドレス, 送信元論理アドレス, 送信先論理アドレスの組み合わせで設定される。論理リンクは, 双方向の通信として利用可能である。バルクメッセージで利用されるチャンネルも, 送信元物理アドレス, 送信先物理アドレス, 送信元論理アドレス, 送信先論理アドレスの組み合わせで設定され, 送信元から送信先への片方向のメッセージ送信となる。MACタイプメッセージは, バスリセット, デバイスリセット, 物理ユニット接続, 物理ユニット離脱, ステータス要求, などを定義している。2つの論理ユニット間で, 複数の種類のコネクションを同時に設定することも可能である。

4.3 データリンクフレームフォーマット

図 5にCAN 2.0B 拡張フレームフォーマットを示す。このCANのフレームフォーマット上に図 3で示したデータリンク層のフレームフォーマットをマッピングするが, CANのフレームフォーマットのうち, ここで利用するのは, 11ビットの Identifier, 18ビットの拡張 Identifier, データ長を示す4ビットのDLCのフィールドである。0から64ビットのDataフィールドはデータ送信として利用する。これら以外のフィールドは, CANの仕様で設定されている定義から変更していない。

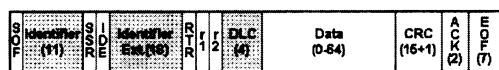


図 5: CAN 2.0B フレームフォーマット

データリンク層フレームフォーマットのCAN 2.0Bフレームフォーマットへのマッピングを表 1に示す。この表において, "1"はCANのアービトレーションビットのドミナントを, "0"はレセシブを示す。

表 1: データリンクの CAN2.0B へのマッピング

Type	CAN ID (18)					MAC Data	
	17 - 14	13	12	11 - 8	7 - 0		
MAC	Cmd Type	Reserve			MAC Data		
Single	Src Logical Addr	Dest Logical Addr			Data		
Block	Logical Link ID						
Bulk	Channel ID		Msg High Count				
Reserve (Stream)	1111	Stream ID		Msg High Count			

Type	CAN ID (11)					Dest Phy Addr
	28	27	26	25 - 22	21 - 18	
MAC	0	0	Bcast/P2PP	Src Phy Addr		Dest Phy Addr
Single	0	1		Src Phy Addr		
Block	1	0	1	Priority		
Bulk	1	1	1	Priority		
Reserve (Stream)	0	0	1	Src Phy Addr		

Type	CAN DLC	
	3	2-0
MAC	0	0
Single	1/0	DLC
Block	1	Msg Count
	1/0	DLC
Bulk	1	Msg Count
	1/0	DLC
Reserve (Stream)	1	Msg Low Count
	1/0	DLC

第28, 27ビットにおいて, MAC, シングル, ブロック, バルクの4タイプのメッセージの分類を行う。MACタイプメッセージで, コマンドタイプ(Cmd Type)が1111の設定は, 64kbps程度の帯域を持つ音声ストリームデータの伝送をサポートするため, リザーブとしている。MACおよびシングルタイプは, ポイント・トゥ・ポイントとブロードキャストがあり, これはBcast/P2PPの第26ビットで示す。

論理ノードアドレスを5ビットで示すように, 物理ユニットあたり最大設置ノード数(実行アプリケーション数)は32となる。バルクタイプメッセージのカウンタは, 設定値が大きいほど効率が良い。ここでは, カウンタフィールドを2つにわけ, 上位を第11から第8ビットで示し, 下位をDLC内で示す。DLCは通常, CANのデータ長を示すが, DLCの第3ビットが1の場合は, 8バイトデータ長を示し, その場合に限り, 第2から第0ビットは任意の値となる。この任意の値をカウンタフィールドとして利用する。メッセージ中の最後のフレームのみ8バイト以下のデータ長となるため, その場合はデータ長を示す。

5. トランスポート層

論理的通信モデルを実現するにあたり、データリンク層と同様に、トランスポート層も重要な役割を果たす。ここでは、エラーチェックおよび再送機能を含むエンド・トゥ・エンドのデータ到達確認を行い、分割されて送信されたメッセージをまとめ、アプリケーション層に対して適切な形式でデータを伝える。

5.1 トランスポートフレームフォーマット

図6にトランスポート層のフレームフォーマットの基本構成を示す。コマンドとデータの2種類のフォーマットがある。コマンド、データ共に、コントロール部である1バイトのヘッダがある。この1バイトのヘッダは、LLCフォーマットのCAN ID フィールドの第7から第0ビット(Data)に格納されるため、効率よく伝送することが可能となっている。

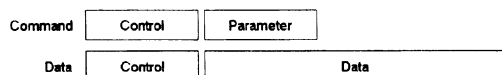


図6: トランスポート層フレームフォーマット

コマンドは、シングルタイプメッセージのためのACK、ブロックタイプメッセージおよびブロックタイプメッセージのためのACK、また、ブロックタイプメッセージおよびバルクタイプメッセージのためのNACKがある。これらのACK/NACKメッセージは、LLCのシングルタイプメッセージとして送信される。ブロック/バルクタイプメッセージのコネクションの設定もコマンドとして送信される。

ヘッダ部には、4ビット長のパケットカウンタがあり、分割されたメッセージのシーケンス番号を入れる。ブロックタイプメッセージのPDUのサイズは最大1K (64 × 16) バイトであり、バルクタイプメッセージは最大16K (1024 × 16) バイトとなる。

6. その他の階層

本研究では、アプリケーション層におけるメッセージのフレームフォーマットについて、特に規定していない。ここでは、アプリケーション層のメッセージ長は、トランスポート層の最大値と設定して、ブロックタイプメッセージの場合1kバイト、バルクタイプメッセージは16kバイトとしている。これ以上のサイズのデータの場合は、アプリケーションにおいて分割して送信する必要がある。

物理層に関しては、CANの仕様から変更する点はないが、プラグ・アンド・プレイを実現するためには、機器接続時に、IDB-Cで規定しているID割り当てのための追加仕様が必要となる。また、IDB-Cとする場合、最大伝送速度は250Kbpsとなる。

7. 実装と評価

7.1 プロトコル実装

本研究で検討した論理的通信モデルを実現するための優先度制御を行うデータ転送プロトコルの実装を行った。下位ネットワークはCAN 2.0Bとし、マイクロプロセッサとしてCANコントローラを内蔵している三菱電機(ルネサス・テクノロジー)製のM16C (16MHz)を利用した。CANの速度は、IDB-C物理層仕様と比較するために、250Kbpsと設定した。

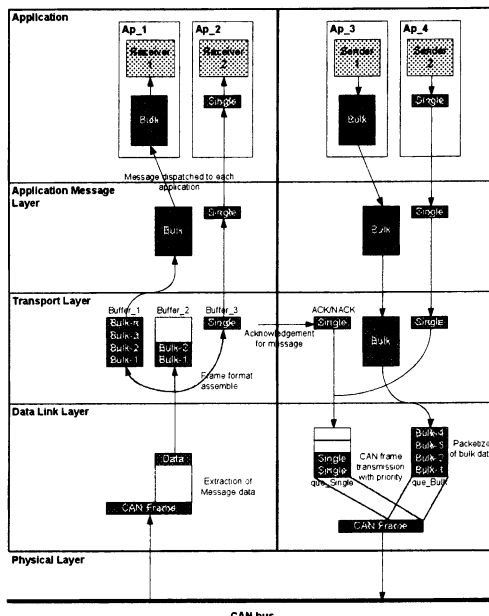


図7: プロトコル実装概要

図7に本プロトコル実装の概要を示す。プロセッサ内のCANコントローラを制御するデバイスドライバの上に、データリンク層、トランスポート層、アプリケーション層があり、その上位にアプリケーションを搭載している。1つのノードにシングルタイプメッセージ送信アプリケーション、バルクタイプメッセージ送信アプリケーションを搭載し、他のノードに、シングルタイプメッセージ受信アプリケーション、バルクタイプメッセージ受信アプリケーションを搭載した。この2つのノードをCANネットワーク経由で接続し、データの送受信テストを行った。メッセージの送受信が正常に行われるかはもちろんであるが、ここでは、メッセージの優先度が認識され、優先度の低いメッセージを送信中に、優先度の高いメッセージが遅れなく到達するかを確認することに主眼を置いた。性能を評価するために、内部に10μsのソフトウェアタイマーを設定し、これに

より時間の計測を行う。

7.2 性能評価

実装したネットワークプロトコルについての優先度に基づいたメッセージ伝送の評価を行うため、シングルタイプメッセージ伝送、バルクタイプメッセージ伝送それぞれの個別送信、およびバルクタイプメッセージの送信中にシングルタイプメッセージ伝送する複合送信の時間を計測した。評価条件として、シングルタイプメッセージは最長の8バイトとし、バルクタイプメッセージは、送信中にシングルタイプを送るために十分な長さの1kバイトと設定した。計測する送信時間は、アプリケーションがメッセージ送信のための関数と呼んだ時点から、メッセージ送信完了の割り込み関数が完了するまでを計測している。システムとして組み込んだ10 μ sのソフトウェアタイマーで計測するとともに、Vector社のCANプロトコルアナライザであるCANoe[7]を利用して、伝送されるフレームの計測も実施した。

表 2: メッセージの送信時間

	送信時間(平均)	
	シングルタイプ メッセージ	バルクタイプ メッセージ
個別送信	1.59 ms	106.1 ms
複合送信	1.71 ms	107.4 ms

表 2にそれぞれのメッセージの送信時間の測定結果を示す。シングルタイプメッセージの個別の送信時間は平均1.59 msで、その中で、アプリケーションが送信のための関数呼び出ししてからデータリンク層がデータを送出するまでの時間が約1 ms、1CANフレームがバスに完全に送られるまでが約670 μ s. かかっていることが計測された。バルクタイプメッセージの個別の送信時間は平均106.1 msで、その中で、アプリケーション層からデータリンク層までメッセージが送られる時間はおよそ20 msである。バルクタイプメッセージ送信中にシングルタイプメッセージを送信する複合送信では、シングルタイプメッセージの平均の送信時間は1.71 msであり、バルクタイプメッセージの平均の送信時間は107.4 msとなった。

1つのCANのフレームをネットワーク上で伝送するために必要な時間は512 μ s (128 bits/250 kbps)であり、次のCANフレームは前のCANフレームの伝送が完了した後に送信される。バルクタイプメッセージ送信中にシングルタイプメッセージを送信するためには、最大でこれだけの遅延が発生する場合がある。実際、シングルタイプメッセージを送出するまでの遅延時間を計測すると約120 μ sであり、このことから、ここでのネットワークプロトコルは、優先度に基づいたメッセー

ジ伝送において正常に機能していることがわかる。バルクメッセージの送信時間は、個別送信より複合送信のほうが約1.3 ms長い。これはシングルタイプメッセージの送信フレームとACKフレームの伝送に必要なバス占有時間が約1.2 msであり、それに追加して、ACKフレーム受信処理時間がおおよそ300 μ s必要となり、この計測された遅延時間はほぼ妥当な値と考えられる。また、CANoeを利用して、メッセージ送受信の経緯を測定すると、バルクタイプメッセージ送信中にシングルタイプメッセージが送信されており、本ネットワークプロトコルで設計した優先度に基づいたメッセージ伝送が正常に動作していることが確認できた。

8. まとめ

本研究では、論理的通信モデルの検討を行い、CAN上で実現できる優先度制御可能なアルタイムデータ伝送のためのデータリンク層、トランスポート層のプロトコルを開発した。CAN (2.0B)メッセージは、29ビットのIDと最大8バイトのデータフレームから構成されており、上位に通信モデルを構築するには制限が大きい。そのため、本論文では、各階層のプロトコルのフレームフォーマットを、効率的にCANフレームにマッピングする方法についても具体的に記述した。そして、このプロトコルを、CANコントローラ内蔵のM16Cマイコンに実装し、評価を行った。その結果、IDB-Cにおいて実現できていない論理的通信モデルの構築に加え、応答時間、データ転送効率、CPU負荷についての問題について改善していることを立証することができた。

文 献

- [1] Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling, ISO 11898-1, 2003.
- [2] Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit, ISO 11898-2, 2003.
- [3] ITS Data Bus – IDB-C Physical Layer (J2366-1), Society of Automotive Engineering, 2001.
- [4] ITS Data Bus – IDB-C Data Link Layer (J2366-2), Society of Automotive Engineering, 2001.
- [5] ITS Data Bus – IDB-C Thin Transport Layer (J2366-4), Society of Automotive Engineering, 2002.
- [6] ITS Data Bus – IDB-C Application Message Layer (J2366-7), Society of Automotive Engineering, 2002.
- [7] Vector Informatik Web Page:
<http://www.vector-informatik.de/english/products/buses.php>